

IN THE CLAIMS:

Please substitute the following claims for the same-numbered claims in the application:

1. (Currently Amended) A method for compiling Unified Parallel C-language (UPC) source code containing UPC-unique constructs and C-language constructs, the said method comprising the steps of:

translating said UPC source code into a first intermediate form;

generating proxy form C-language code strings of data components within said first intermediate form UPC unique constructs;

inserting said generated code strings into said UPC source code to form a combined code;

translating said combined code into a second intermediate form, wherein any statements within references to said UPC-unique constructs are placed in a C-form with replaced by references to corresponding proxy components in associated program text, and surviving UPC-unique constructs are discarded; and

converting said second intermediate form to compiled machine code.

2. (Original) The method of claim 1, wherein said code strings are proxy declarations.

3. (Previously Presented) The method of claim 2, wherein each said proxy declaration includes a name that is a mangled version of a name of a respective UPC-unique data component having a one-to-one mapping.

4. (Original) The method of claim 1, wherein said associated program text includes a conditional statement.
5. (Previously Presented) The method of claim 4, wherein said UPC-unique constructs are forall statements, and said associated program text includes a conditional statement whose predicates leads to evaluation based upon an affinity test.
6. (Currently Amended) The method of claim 5, wherein, for forall statements having an affinity affinity other than continue, the translation step includes sub-traversal of a forall body and determining the context of each static level of nesting.
7. (Currently Amended) The method of claim 6, further comprising the step of incrementing a depth variable in accordance with each said sub-traversal.
8. (Currently Amended) A method for compiling Unified Parallel C-language (UPC) UPC source code, said method comprising the steps of:
 - (a) —converting UPC-unique constructs into C-level form;
 - (b) —inserting said C-level constructs into said source code to form a combined code;
 - (c) —translating said combined code to an intermediate form, wherein any surviving UPC-unique components are discarded; and
 - (d) —converting said intermediate form to compiled machine code.

9. (Original) The method of claim 8, wherein said C-level form constructs are in a form having proxy declarations.

10. (Previously Presented) The method of claim 9, wherein each said proxy declaration for a UPC-unique data construct includes a name that is a mangled version of a name of the respective UPC-unique data component having a one-to-one mapping.

11. (Previously Presented) The method of claim 9, wherein said proxy declaration for a UPC-unique statement includes a conditional statement.

12. (Original) The method of claim 11, wherein, for forall statements, said conditional statement has predicates leading to evaluation based upon an affinity test.

13. (Currently Amended) A method for compiling Unified Parallel C language (UPC) source code containing UPC unique constructs and C language constructs, the method comprising the steps of:

translating said UPC source code into a first intermediate form, including any UPC unique statements being placed in a C form with associated program text;

generating proxy form C language code strings of UPC unique data components within said first intermediate form;

inserting said generated code strings into said UPC source code to form a combined code;

~~translating said combined code and said C form statements and associated program text to a second intermediate form, wherein surviving UPC unique components are discarded; and converting said second intermediate form to compiled machine code.~~

A program storage device readable by computer, tangibly embodying a program of instructions executable by said computer to perform a method for compiling Unified Parallel C-language (UPC) source code containing UPC-unique constructs and C-language constructs, said method comprising:

translating said UPC source code into a first intermediate form;
generating proxy form C-language code strings of data components within said first intermediate form;

inserting said generated code strings into said UPC source code to form a combined code;
translating said combined code into a second intermediate form, wherein any references to said UPC-unique constructs are replaced by references to corresponding proxy components in associated program text, and surviving UPC-unique constructs are discarded; and
converting said second intermediate form to compiled machine code.

14. (Currently Amended) The ~~method~~ program storage device of claim 13, wherein said code strings are proxy declarations.

15. (Currently Amended) The ~~method~~ program storage device of claim 14, wherein each said proxy declaration includes a name that is a mangled version of a name of a respective UPC-unique data component having a one-to-one mapping.

16. (Currently Amended) The method program storage device of claim 13, wherein said associated program text includes a conditional statement.
17. (Currently Amended) The method program storage device of claim 16, wherein said UPC-unique constructs are forall statements, and said associated program text includes a conditional statement whose predicates leads to evaluation based upon an affinity test.
18. (Currently Amended) The method program storage device of claim 17, wherein, for forall statements having an affinity affinity other than continue, the translation step includes sub-traversal of a forall body and determining the context of each static level of nesting.
19. (Currently Amended) The method program storage device of claim 18, wherein said method further ~~comprising the step of~~ comprises incrementing a depth variable in accordance with each said sub-traversal.
20. (Currently Amended) A Unified Parallel C-language (UPC) UPC compiler comprising:
 - a front end module operable for receiving UPC source code;
 - an intermediate form processor operable for converting UPC-unique constructs into C-level form;
 - a feedback loop to said front end processor,wherein said front end ~~processor~~ module further inserts said C-level constructs into said

source code to form a combined code;

wherein said intermediate form processor translates said combined code to an intermediate form, wherein any surviving UPC-unique components are discarded; and a back end module operable for converting said intermediate form to compiled machine code.

21. (Original) The compiler of claim 20, wherein said C-level form constructs are in a form having proxy declarations.

22. (Previously Presented) The compiler of claim 21, wherein each said proxy declaration for a UPC-unique data construct includes a name that is a mangled version of a name of a respective UPC-unique data component having a one-to-one mapping.

23. (Original) The compiler of claim 21, wherein said proxy declaration for a UPC-unique statement includes a conditional statement.

24. (Original) The compiler of claim 23, wherein, for forall statements, said conditional statement has predicates leading to evaluation based upon an affinity test.

25. (Currently Amended) The compiler of claim 24, wherein, for forall statements having an affinity affinity other than continue, the intermediate form ~~processor translates processor~~ translates by sub-traversal of a forall body and determination of the context of each static level of

nesting.

26. (Original) The compiler of claim 25, wherein the intermediate form processor increments a depth variable in accordance with each said sub-traversal.

27-28. (Cancelled).

29. (Currently Amended) A program storage device readable by computer, tangibly embodying a program of instructions executable by said computer to perform a method for compiling Unified Parallel C-language (UPC) UPC source code, said method comprising:

converting UPC-unique constructs into C-level form;
inserting said C-level constructs into said source code to form a combined code;
translating said combined code to an intermediate form, wherein any surviving UPC-unique components are discarded; and
converting said intermediate form to compiled machine code.

30. (Previously Presented) The program storage device of claim 29, wherein said C-level form constructs are in a form having proxy declarations.

31. (Previously Presented) The program storage device of claim 30, wherein each said proxy declaration for a UPC-unique data construct includes a name that is a mangled version of a name of the respective UPC-unique data component having a one-to-one mapping.

32. (Previously Presented) The program storage device of claim 30, wherein said proxy declaration for a UPC-unique statement includes a conditional statement.
33. (Previously Presented) The program storage device of claim 32, wherein, for forall statements, said conditional statement has predicates leading to evaluation based upon an affinity test.
34. (Currently Amended) A parallel distributed shared memory computer system having a single real address space, comprising:
 - a plurality of processor modules;
 - a memory unit associated with each processor module; and
 - an interconnection network linking all processor modules and memory units;and wherein each processor module includes a Unified Parallel C-language (UPC) UPC compiler module, each said UPC compiler module including:
 - a front end module operable for receiving UPC source code;
 - an intermediate form processor operable for converting UPC-unique constructs into C-level form;
 - a feedback loop to said front end processor,
wherein said front end ~~processor~~ module further inserts said C-level constructs into said source code to form a combined code;
 - wherein said intermediate form processor translates said combined code to an

intermediate form, wherein any surviving UPC-unique components are discarded; and
a back end module operable for converting said intermediate form to compiled
machine code.

35. (Previously Presented) The computer system of claim 34, wherein said C-level form
constructs are in a form having proxy declarations.

36. (Previously Presented) The computer system of claim 35, wherein each said proxy
declaration for a UPC-unique data construct includes a name that is a mangled version of a name
of the respective UPC-unique data component having a one-to-one mapping.

37. (Previously Presented) The computer system of claim 35, wherein said proxy declaration
for a UPC-unique statement includes a conditional statement.

38. (Previously Presented) The computer system of claim 37, wherein, for forall statements,
said conditional statement has predicates leading to evaluation based upon an affinity test.